



Requirements Peer Review Checklist

Number: 580-CK-057-01
Effective Date: April 24, 2006
Expiration Date: April 24, 2011

Approved By: (signature)
Name: Barbara Pfarr
Title: Assoc. Chief, ISD

Responsible Office: 580/Information Systems Division (ISD)
Title: Requirements Peer Review Checklist

Asset Type: Checklist
PAL Number: 2.2.1.5

Requirements Peer Review Checklist

The Requirements Peer Review Checklist defines the criteria to be used during a peer review of a software requirements specification. For a detailed explanation of how peer reviews are conducted, and how they differ from formal reviews, inspections, and walkthroughs, please refer to "Inspections, Peer Reviews, and Walkthroughs," PAL #3.2.3.

This checklist may be used for all software or system requirements specifications, both new and revised. Software projects and ISD Branches are encouraged to develop and use tailored versions of this checklist.

For each checklist item below, place a check (✓) in the box if the checklist item is satisfied. Otherwise, list any problem areas or exceptions under "Issues and Comments."

		✓	Issues and Comments
1	<p>Compliance with standards – Does the requirements specification comply with ISD or tailored Branch/project-level standards and naming conventions?</p> <p><i>GUIDANCE: If, for example, applicable standards specify that certain material should be included in the requirements specification, and this material is missing, without any explanation, then this should be noted under "Issues and Comments."</i></p> <p><i>GUIDANCE: If any waivers to applicable standards have been granted, make a notation to this effect under "Issues and Comments." Approved ISD standards may be found on the EPG web site at http://software.gsfc.nasa.gov/process.cfm.</i></p>	<input type="checkbox"/>	
2	<p>Completeness of Specifications – Does the requirements specification document address all known requirements? Have 'TBD' requirements been kept to a minimum, or eliminated entirely?</p> <p><i>GUIDANCE: A requirements specification should address such elements as control flow, data transformations, design constraints, and user interface.</i></p>	<input type="checkbox"/>	

3	<p>Clarity – Are the requirements clear enough to be turned over to an independent group for implementation?</p>	<input type="checkbox"/>	
4	<p>Consistency – Are the specifications consistent in notation, terminology, and level of functionality? Are any required algorithms mutually compatible?</p>	<input type="checkbox"/>	
5	<p>External Interfaces – Have external interfaces been adequately defined?</p> <p><i>GUIDANCE: Interface requirements are frequently documented in a separate Interface Requirements Document (IRD) or Interface Control Document (ICD).</i></p>	<input type="checkbox"/>	
6	<p>Testability – Are the requirements testable? Will the testers be able to determine whether each requirement has been satisfied?</p> <p><i>GUIDANCE: The requirements specification should state how every requirement will be tested. This helps to reduce ambiguity, increase clarity, and show testability.</i></p>	<input type="checkbox"/>	
7	<p>Design-Neutrality – Does the requirements specification state what actions are to be performed, rather than how these actions will be performed?</p> <p><i>GUIDANCE: In other words, the requirements should concentrate on what the software needs to do, rather than how it will do it.</i></p> <p><i>GUIDANCE: In the case where a system or subsystem is being configured from a product line, design neutrality does not apply. Instead, one should show that requirements are consistent with the selected product line architecture.</i></p>	<input type="checkbox"/>	
8	<p>Readability – Does the requirements specification use the language of the intended testers and users of the system, not software jargon?</p>	<input type="checkbox"/>	
9	<p>Level of Detail – Are the requirements at a fairly consistent level of detail? Should any particular requirement be specified in more detail? In less detail?</p> <p><i>GUIDANCE: At GSFC, there are at least three levels of requirements. Level 1 is for Mission-level or Project-level requirements, Level 2 for requirements at the software system level, and Level 3 for subsystem-level requirements. Frequently there is also a Level 4, which contains internal, or all-software, requirements. There is normally a separate requirements specification for each level of requirements. It is important that each requirement be stated at an appropriate level of detail, and that all the requirements in a given requirements specification be at the same level of detail.</i></p>	<input type="checkbox"/>	

10	Requirements Singularity – Does each requirement address a single concept, topic, element, or value? <i>GUIDANCE: Avoid compound requirements that do not clearly delineate the parts with separate identifiers.</i>	<input type="checkbox"/>	
11	Definition of Inputs and Outputs – Have the internal interfaces, i.e., the required inputs to and outputs from the software system, been fully defined? Have the required data transformations been adequately specified? <i>GUIDANCE: Note that use of correct units is a commonly occurring issue for data interfaces and transformations.</i>	<input type="checkbox"/>	
12	Scope – Does the requirements specification adequately define boundaries for the scope of the target software system? Are any essential requirements missing?	<input type="checkbox"/>	
13	Design Constraints – Are all stated design and performance constraints realistic and justifiable? <i>GUIDANCE: An example of an unrealistic constraint might be 100% availability of the system, or 1 nanosecond response to the user. Actually, a 1 nanosecond response time might seem unrealistic, but could also be necessary.</i>	<input type="checkbox"/>	
14	Traceability – Has a bidirectional traceability matrix been provided?	<input type="checkbox"/>	
Notes/Action Items for follow-up			
#	Action	Assignee	Due Date

References

- **Glossary:** <http://software.gsfc.nasa.gov/glossary.cfm>. Defines common terms used in ISD process assets
- **Process Asset Library:** <http://software/gsfsc.nasa.gov/process.cfm> Library of all ISD process descriptions
- Adaptable Process Model, System Level Requirements Validation, R. S. Pressman and Associates: <http://www.rspa.com/checklists/sysreqval.html>
- Adaptable Process Model, Software Requirements Specification, R. S. Pressman and Associates: <http://www.rspa.com/checklists/reqspec.html>
- Software Formal Inspections Guidebook, NASA-GB-A302, August 1993, Appendix A, Checklist SY, Functional Requirements Checklist (JPL), pp. 65-66: <http://satc.gsfc.nasa.gov/Documents/fi/gdb/fi.pdf>

Check the Process Asset Library at <http://software.gsfc.nasa.gov/process.cfm> to obtain the latest version.
NOTE: Words or phrases shown in blue underlined contain links to additional information.
Guidance & tailoring information is shown in *italics with gray background*.

-
- Recommended Approach to Software Development, Revision 3, SEL-81-305, Goddard Space Flight Center, June 1992.
 - SSDM Standards and Procedures, Computer Sciences Corporation, 1993.

**Change
History**

Version	Date	Description of Improvements
1.0	4/24/06	Initial approved version by CCB